1ère spé *Notre Dame de La Merci* **Janvier 2024**

Contrôle de Mathématiques sur les suites numériques et python

Une gamme de piano correspond à une suite de notes correspondant à un même mode, soit mineur soit majeur. Une octave commence par la dernière note de la gamme. Celle-ci est inférieure si la gamme de piano est descendante et supérieure, si celle-ci est montante. Autrement dit, une gamme de piano se définit en une suite ordonnée de différents niveaux de tonalités.

Première partie

Exercice 1 Etudier le sens de variation des suites (u_n) définies ci-dessous : *(9 points)*

a) $u_n = -n^2 + 8n + 6 \ (n \in \mathbb{N})$ (sans utiliser de fonction associée)

b) $u_n = \dfrac{2n+4}{5n+9} \ (n \in \mathbb{N})$ (sans utiliser de fonction associée)

c) $u_n = \dfrac{0,7^n}{20n} \ (n \in \mathbb{N})$

Deuxième partie sur feuille en utilisant uniquement votre ordinateur

Exercice 2 *(3 points)*

On considère une suite u définie pour tout $n \in \mathbb{N}$.
On admet que la suite u est strictement croissante.

1) Quelle est la suite étudiée dans le programme ci-contre ?
2) Que fait le programme ci-contre ?
3) En utilisant votre calculatrice, déterminer par la méthode qui vous plaira les valeurs renvoyées par ce programme.

```
u = 1
n = 0
while u < 4.999:
   u = 0.2*u + 4
   n += 1
print("le rang cherché est :" , n)
print("la valeur du rang cherché est :" , u)
```

Exercice 3 *(3 points)*

On considère la suite (u_n) définie par récurrence par : $\begin{cases} u_0 = 100 \\ u_{n+1} = 0,8u_n + 2 \end{cases}$.

On admet que la suite (u_n) est strictement décroissante pour tout $n \in \mathbb{N}$.

Ecrire un programme qui détermine à partir de quel rang tous les termes de la suite deviennent inférieurs à $10,001$.

Exercice 4 *(3 points)*

On considère la suite (u_n) définie par récurrence par : $\begin{cases} u_1 = 3 \\ u_{n+1} = 2u_n + 3 \end{cases}$.

Ecrire un programme qui calcule les 10 premiers termes de cette liste et les sauvegarde dans une liste que l'on affichera à la fin du programme.

Exercice 5 : *(2 points)*

Ecrire un programme permettant de calculer la somme :

$$\frac{5}{1^3} + \frac{5}{2^3} + \frac{5}{3^3} + \dots + \frac{5}{1000^3}$$

Donner une valeur approchée avec 6 décimales.

Contrôle de Mathématiques sur les suites numériques – CORRIGE – M. Quet

Première partie

Exercice 1 *Etudier le sens de variation des suites (u_n) définies ci-dessous :*

a) $u_n = -n^2 + 8n + 6 \quad (n \in \mathbb{N})$ (sans utiliser de fonction associée)

$$u_{n+1} - u_n = \left(-(n+1)^2 + 8(n+1) + 6\right) - \left(-n^2 + 8n + 6\right)$$
$$= -n^2 - 2n - 1 + 8n + 8 + 6 + n^2 - 8n - 6$$
$$= -2n + 7$$

$$u_{n+1} - u_n < 0 \iff -2n + 7 < 0 \iff -2n < -7 \iff \frac{-2n}{-2} > \frac{-7}{-2} \iff n > \frac{7}{2}$$

Donc pour $n \geq 4$: la suite (u_n) est strictement décroissante.

NB : certains ont résonné autrement en écrivant :

$$u_{n+1} - u_n > 0 \iff -2n + 7 > 0 \iff -2n > -7 \iff \frac{-2n}{-2} < \frac{-7}{-2} \iff n < \frac{7}{2}$$

→cela signifie que la suite (u_n) est strictement croissante pour $n < 4$ seulement.

b) $u_n = \dfrac{2n+4}{5n+9} \quad (n \in \mathbb{N})$ (sans utiliser de fonction associée)

$$u_{n+1} - u_n = \frac{2(n+1)+4}{5(n+1)+9} - \frac{2n+4}{5n+9} = \frac{2n+2+4}{5n+5+9} - \frac{2n+4}{5n+9} = \frac{2n+6}{5n+14} - \frac{2n+4}{5n+9}$$

$$= \frac{2n+6}{5n+14} \times \frac{5n+9}{5n+9} - \frac{2n+4}{5n+9} \times \frac{5n+14}{5n+14} = \frac{10n^2 + 18n + 30n + 54}{(5n+14)(5n+9)} - \frac{10n^2 + 28n + 20n + 56}{(5n+14)(5n+9)}$$

$$= \frac{\left(10n^2 + 18n + 30n + 54\right) - \left(10n^2 + 28n + 20n + 56\right)}{(5n+14)(5n+9)}$$

$$= \frac{10n^2 + 18n + 30n + 54 - 10n^2 - 28n - 20n - 56}{(5n+14)(5n+9)} = \frac{-2}{(5n+14)(5n+9)}$$

$n \in \mathbb{N}$ donc : $5n + 14 > 0$ et $5n + 9 > 0$

Ainsi : $u_{n+1} - u_n < 0$ et la suite (u_n) est strictement décroissante pour tout $n \in \mathbb{N}$.

c) $u_n = \dfrac{0,7^n}{20n} \quad (n \in \mathbb{N}^*)$

$$\frac{u_{n+1}}{u_n} = \frac{\dfrac{0,7^{n+1}}{20(n+1)}}{\dfrac{0,7^n}{20n}} = \frac{0,7^{n+1}}{20(n+1)} \times \frac{20n}{0,7^n} = \frac{20n}{20(n+1)} \times \frac{0,7^{n+1}}{0,7^n} = \frac{n}{n+1} \times \frac{0,7 \times \boxed{0,7^n}}{\boxed{0,7^n}} = \frac{0,7n}{n+1}$$

$n \in \mathbb{N}^*$ donc :

$$0,7 < 1 \iff 0,7n < n < n+1 \iff \frac{0,7n}{n+1} < \frac{n+1}{n+1} \iff \frac{u_{n+1}}{u_n} < 1$$

$$n < n+1 \iff \frac{n}{n+1} < 1 \iff \frac{n}{n+1} \times 0,7 < 0,7 < 1$$

Tous les termes de la suite (u_n) sont positifs, donc la suite (u_n) est strictement décroissante pour tout $n \in \mathbb{N}^*$.

Deuxième partie sur feuille en utilisant votre ordinateur

Exercice 2

On considère une suite u définie pour tout $n \in \mathbb{N}$.
On admet que la suite u est strictement croissante.

```
u = 1
n = 0
while u < 4.999:
   u = 0.2*u + 4
   n += 1
print("le rang cherché est :" , n)
print("la valeur du rang cherché est :" , u)
```

1) *Quelle est la suite étudiée dans le programme ci-contre ?*
   La suite u est définie pour tout $n \in \mathbb{N}$ par :
   $u_0 = 1$ et $u_{n+1} = 0,2u_n + 4$.
2) *Que fait le programme ci-contre ?*
   Ce programme cherche le premier rang à partir duquel cette suite *u* croissante dépasse la valeur 4,999.
   Elle restitue également la première valeur de la suite dépassant 4,999.
3) *Exécuter ce programme et déterminer les valeurs renvoyées par ce programme.*
   On obtient :
   le rang cherché est : 6
   la valeur du rang cherché est : 4.999744

Exercice 3

On considère la suite (u_n) définie par récurrence par : $\begin{cases} u_0 = 100 \\ u_{n+1} = 0,8u_n + 2 \end{cases}$.

On admet que la suite (u_n) est strictement décroissante pour tout $n \in \mathbb{N}$.

Ecrire un programme qui détermine à partir de quel rang tous les termes de la suite deviennent inférieurs à $10,001$.

```
u = 100
n = 0
while u >= 10.001:
   u = 0.8*u + 2
   n += 1
print("le rang cherché est :" , n)
print("la valeur du rang cherché est :" , u)
```

On obtient :
le rang cherché est : 52
la valeur du rang cherché est : 10.000822094671001

Exercice 4

On considère la suite (u_n) définie par récurrence par : $\begin{cases} u_1 = 3 \\ u_{n+1} = 2u_n + 3 \end{cases}$.

Ecrire un programme qui calcule les 10 premiers termes de cette liste et les sauvegarde dans une liste que l'on affichera à la fin du programme.

```
u = 3
n = 1
liste = [3]
for i in range(2,11):
   u = 2*u + 3
   n += 1
   liste.append(u)
print(liste)
```

On obtient :
[3, 9, 21, 45, 93, 189, 381, 765, 1533, 3069]

Exercice 5 :

Ecrire un programme permettant de calculer la somme :

$$\frac{5}{1^3}+\frac{5}{2^3}+\frac{5}{3^3}+...+\frac{5}{1000^3}$$

Donner une valeur approchée avec 6 décimales.

```
S = 0
for i in range(1,1001):
   S += 5/i**3
print(S)
```

On obtient :

6.010282018296714